

#4



Our Docket No: 42390.P9020

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)	
)	
Simon Knee, et al.)	Examiner: Unassigned
)	
Application No: 09/823,616)	Art Unit: Unassigned
)	
Filed: March 31, 2001)	
)	
For: FAST CLASSLESS INTER-DOMAIN)	
ROUTING (CIDR) LOOKUPS)	
)	

Preliminary Amendment

Box Fee Amendment
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Prior to examination of the above-captioned case, the Applicant respectfully requests the Examiner to enter the following amendment and to consider the following

remark:

06/28/2001 KZEWDIE 00000060 09823616

01 FC:103
02 FC:102

216.00 GP
328.00 GP

FIRST CLASS CERTIFICATE OF MAILING

I hereby certify that I am causing the above-referenced correspondence to be deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

June 20, 2001

Date of Deposit

Fran C. Rolfsen

Name of Person Mailing Correspondence

Fran C. Rolfsen
Signature

6-20-01
Date

AMENDMENTS -- CLEAN VERSION

In the Specification:

Please replace paragraph 0027 on page 10 with the following paragraph:

[0027] Generally speaking, the routing table 230 includes information that indicates on which port (the egress port) a particular address is located. In addition to the specific egress port through which to send the packet, the table might also store information about how the packet should be modified before forwarding. Such changes might include, but not be limited to, a new Media Access Control (MAC) destination address and Virtual LAN (VLAN) address. The routing table 230 typically includes entries having address data and a corresponding payload. The payload may be the forwarding instructions for the particular address or an indication (e.g., an index or a pointer) that can be used to find the forwarding instructions for the particular address. According to one embodiment, the routing table 230 is implemented in whole or part as a Content Addressable Memory (CAM), a random access memory (RAM), such as synchronous RAM (SRAM), or the like that may be implemented using hashing techniques. For example, a hash index generator (not shown), such as a cyclic redundancy checksum (CRC) generator may produce a hash index based on the destination IP address of the packet to be forwarded and the corresponding mask.

Please replace paragraph 0029 on page 11 with the following paragraph:

[0029] Briefly, in operation, the routing process 220 determines appropriate masks to apply to a search key associated with a received packet with reference to the mask table 240 and then queries the routing table 230 to identify the appropriate

forwarding instructions for the received packet. For example, the routing process 220 may retrieve an encoded mask vector corresponding to a destination network layer address contained in the received packet (e.g., a source or destination IP address) from the mask table 240 and then perform one or more address look-up requests using those of the masks indicated by the encoded mask vector to have a potential for matching an entry in the routing table 230.

Please replace paragraph 0033 on page 13 with the following paragraph:

[0033] At any rate, using this simplified example, a longest match search will be illustrated for a destination address 0B.01.02.F0 (the search key). In Internet Protocol (IP), there might be several entries of a routing table that match a particular address. In this example, network prefix 331 of route (entry) 330, network prefix 341 of route (entry) 340, and network prefix 351 of route (entry) 350 all match 0B.01.02.F0. However, to assure proper delivery of the packet, a network device must use the most specific matching entry, i.e., the entry having the longest mask. Importantly, to be considered a matching entry, the address associated with the entry must match a portion of the search key identified by its mask and the entry's mask length must be less than or equal to the search key's mask length.

Please replace paragraph 0051 on page 19 with the following paragraph:

[0051] At decision block 730, a determination is made whether any further masks are to be attempted. If MaskWord is not equal to zero, then at least one more mask remains to be tried and processing continues with processing block 740. However,

if MaskWord is equal to zero, then there are no further entries in the routing table 230 that can match the query at any length and longest match search processing is terminated. According to an alternative embodiment, the test for a zero MaskWord is eliminated and instead a special routing table entry is created to match a query of all zeroes and then terminate the search.

Please replace paragraph 0057 on page 21 with the following paragraphs:

[0057] The longest mask described by a MaskWord is found by forming each bit of the mask from the OR of the corresponding bit and all of the lower bits of the MaskWord.

```
Mask[0] = MaskWord[0];
Mask[1] = | MaskWord[1:0];
Mask[2] = | MaskWord[2:0];
Mask[3] = | MaskWord[3:0];
...
Mask[31] = | MaskWord[31:0];
```

where " $|$ MaskWord[i,0]" is the REDUCTION OR operation which forms the OR of all bits in the identified range.

This approach of forming the mask employs some very large OR gates for the higher numbered bits. To avoid this people sometimes employ a ripple technique by observing that each bit of the mask already includes the OR of the corresponding bit and all of the lower bits of the MaskWord.

```
Mask[0] = MaskWord[0];
Mask[1] = MaskWord[1] || Mask[0];
Mask[2] = MaskWord[2] || Mask[1];
Mask[3] = MaskWord[3] || Mask[2];
...
Mask[31] = MaskWord[31] || Mask[30];
```

This ripple approach, sometimes called the recursive approach, has the advantage of reducing all of the logic to 31 two-input OR gates and the disadvantage of being slower because it takes 31 gate delays to determine the value of the highest order bit.

A look-ahead method compromises between these extremes of wide gates and long delays by using the ripple approach for most of the bits, but using an occasional wider gate to anticipate what the ripple value will be. Bits higher than the wider gates use their value rather than the ripple value.

```
Mask[0] = MaskWord[0];
Mask[1] = MaskWord[1] || Mask[0];
Mask[2] = MaskWord[2] || Mask[1];
Mask[3] = MaskWord[3] || Mask[2];
Mask[4] = MaskWord[4] || Mask[3];
Mask[5] = MaskWord[5] || Mask[4];

Mask[6] = (MaskWord[6:0]) || Mask[5]; // anticipated value

Mask[7] = MaskWord[7] || Mask[6];
Mask[8] = MaskWord[8] || Mask[7];
Mask[9] = MaskWord[9] || Mask[8];
Mask[10] = MaskWord[10] || Mask[9];
Mask[11] = MaskWord[11] || Mask[10];

Mask[12] = (MaskWord[12:7]) || Mask[6]; // anticipated value

Mask[13] = MaskWord[13] || Mask[12];
Mask[14] = MaskWord[14] || Mask[13];
Mask[15] = MaskWord[15] || Mask[14];
...
Mask[31] = MaskWord[31] || Mask[30];
```

As mentioned earlier, those trying to implement this type of mask expansion in software or in FPGAs cannot efficiently use typical approaches available to ASIC designers. However, processors do have efficient instruction sets and many FPGAs have circuitry “tuned” to implement arithmetic functions. Therefore, if the target platform is

software running on a standard processor or an FPGA, the longest mask can be expanded from an encoded mask vector by the following equation, which provides a fast, compact solution:

$$\text{Mask} = (0 - \text{MaskWord}) \mid \text{MaskWord}$$

In the Claims:

Presented below are the claims, as amended, in a clean, unmarked format with changes entered and not marked. For the Examiner's convenience, all pending claims are presented herein. Claims that remain unchanged by this amendment are prefixed with "(Unchanged)."

Please amend claim 9 as follows:

- 1 1. (Unchanged) A method of performing a longest match search comprising:
2 receiving a search key;
3 determining a set of masks that when applied to the search key are known to have
4 a potential for matching an entry in a routing table;
5 forming a routing table query based upon the search key and a longest mask of the
6 set of masks; and
7 applying the routing table query to the routing table.
- 1 2. (Unchanged) The method of claim 1, further comprising:
2 removing the longest mask from the set of masks; and
3 continuing to apply additional routing table queries until either the set of masks is
4 empty or a matching entry is found in the routing table.
- 1 3. (Unchanged) The method of claim 1, wherein the search key comprises an
2 Internet Protocol (IP) address.

- 1 4. (Unchanged) The method of claim 3, wherein the IP address comprises a
2 destination address.
- 1 5. (Unchanged) The method of claim 3, wherein the IP address comprises a source
2 address.
- 1 6. (Unchanged) The method of claim 1, wherein said determining a set of masks
2 comprises retrieving an encoded mask vector from a mask table based upon the
3 search key, the encoded mask vector having N bits and capable of identifying N
4 different length masks.
- 1 7. (Unchanged) The method of claim 1, wherein the longest mask of the set of
2 masks is determined by the following equation: $\text{Mask} = (0 - \text{MaskWord}) \mid$
3 MaskWord ,
4 where:
5 MaskWord is an encoded mask vector, and
6 Mask is the longest mask identified by MaskWord .
- 1 8. (Unchanged) A packet forwarding device comprising:
2 a plurality of ports upon which packets are received and transmitted;
3 a routing processor coupled to the plurality of ports to determine an egress port of
4 the plurality of ports for a packet received on an ingress port of the
5 plurality of ports by performing a longest match search comprising one or
6 more routing table queries;
7 a routing table, coupled to the routing processor, to provide the routing processor
8 with a match indication and information regarding a matching routing
9 table entry, if any, of a plurality of routing table entries stored therein in
10 response to a routing table query; and

11 a mask table, coupled to the routing processor, to maintain encoded mask vectors
12 identifying mask lengths of the plurality of routing table entries.

1 9. (Amended) The packet forwarding device of claim 8, wherein the encoded mask
2 vectors comprise N-bits and are capable of representing N different masks.

1 10. (Unchanged) The packet forwarding device of claim 8, wherein the routing table
2 comprises a Content Addressable Memory (CAM).

1 11. (Unchanged) The packet forwarding device of claim 8, wherein the one or more
2 routing table queries are formed by applying a series of masks determined with
3 reference to the mask table to a search key extracted from the received packet.

1 12. (Unchanged) A method of forwarding a packet comprising:
2 receiving a packet on an ingress port of a plurality of ports;
3 extracting a destination Internet Protocol (IP) address from a header of the packet;
4 using a portion of the destination IP address to index into a mask table to retrieve
5 an encoded mask vector that identifies a series of masks to be applied to
6 the destination IP address during a longest match search of a routing table,
7 the series of masks representing those masks that are known to have a
8 potential for matching an entry in the routing table when applied to the
9 destination IP address;
10 identifying a longest matching entry in the routing table by performing the longest
11 match search based upon the destination IP address and one or more of the
12 series of masks; and
13 forwarding the packet to a network device associated with the destination IP
14 address via an egress port of the plurality of ports identified by the longest
15 matching entry.

- 1 13. (Unchanged) The method of claim 12, wherein the portion of the destination IP
2 address comprises the most significant N bits of the destination IP address.
- 1 14. (Unchanged) The method of claim 12, wherein the encoded mask vector includes
2 a plurality of mask length indicator bits that each indicate a mask length by virtue
3 of their position within the encoded mask vector.
- 1 15. (Unchanged) The method of claim 12, further comprising updating the mask
2 table to include a new encoded mask vector in response to receiving a new routing
3 table entry.
- 1 16. (Unchanged) A machine-readable medium having stored thereon data
2 representing sequences of instructions, the sequences of instructions which, when
3 executed by a processor, cause the processor to:
4 receive a search key;
5 determine a set of masks that when applied to the search key are known to have a
6 potential for matching an entry in a routing table;
7 form a routing table query based upon the search key and a longest mask of the
8 set of masks; and
9 apply the routing table query to the routing table.
- 1 17. (Unchanged) The machine-readable medium of claim 16, wherein the longest
2 mask of the set of masks is determined by the following equation: $\text{Mask} = (0 -$
3 $\text{MaskWord}) \mid \text{MaskWord}$,
4 where:
5 MaskWord is an encoded mask vector, and
6 Mask is the longest mask identified by MaskWord.

1 18. (Unchanged) The machine-readable medium of claim 16, wherein the set of
2 masks is determined by retrieving an encoded mask vector from a mask table
3 based upon the search key, the encoded mask vector having N bits and capable of
4 identifying N different length masks.

Please add the following claims:

1 19. (New) A method of forwarding a packet comprising the steps of:
2 a step for receiving a packet on an ingress port of a plurality of ports;
3 a step for extracting an Internet Protocol (IP) address from a header of the packet;
4 a step for using a portion of the IP address to index into a mask table to retrieve an
5 encoded mask vector that identifies a series of masks to be applied to the
6 IP address during a longest match search of a routing table, the series of
7 masks representing those masks that are known to have a potential for
8 matching an entry in the routing table when applied to the IP address;
9 a step for identifying a longest matching entry in the routing table by performing
10 the longest match search based upon the IP address and one or more of the
11 series of masks; and
12 a step for forwarding the packet to a network device based upon the longest
13 matching entry.

1 20. (New) The method of claim 19, wherein the IP address comprises a destination
2 IP address.

1 21. (New) The method of claim 19, wherein the IP address comprises a source IP
2 address.

1 22. (New) The method of claim 19, wherein the encoded mask vector includes a
2 plurality of mask length indicator bits that each indicate a mask length by virtue
3 of their position within the encoded mask vector.

1 23. (New) A method of performing a longest match search comprising the steps of:
2 a step for receiving a search key;

a determination step for determining a set of masks that when applied to the search key are known to have a potential for matching an entry in a routing table;
a step for forming a routing table query based upon the search key and a longest mask of the set of masks; and
a step for applying the routing table query to the routing table.

24. (New) The method of claim 23, further comprising the steps of:
a step for removing the longest mask from the set of masks; and
a step for continuing to apply additional routing table queries until either the set of masks is empty or a matching entry is found in the routing table.

25. (New) The method of claim 23, wherein the search key comprises an Internet Protocol (IP) address.

26. (New) The method of claim 23, wherein said determination step comprises retrieving an encoded mask vector from a mask table based upon the search key, the encoded mask vector having N bits and capable of identifying N different length masks.

27. (New) The method of claim 1, wherein the longest mask of the set of masks is determined by the following equation: $\text{Mask} = (0 - \text{MaskWord}) \mid \text{MaskWord}$, where:
MaskWord comprises an encoded mask vector, and
Mask comprises the longest mask identified by MaskWord.

28. (New) The method of claim 7, further comprising:
isolating an endbit of the longest mask;

3 combining the longest mask with the inversion of the longest mask left-shifted
4 one position; and
5 forming a subsequent routing table query based on the masked search key left-
6 shifted one position and the endbit.

1 29. (New) A packet forwarding device comprising:

2 a plurality of interface means for receiving and transmitting packets;
3 routing processor means, coupled to the plurality of interface means, for
4 determining an egress interface of the plurality of interface means for a
5 packet received on an ingress interface of the plurality of interface means
6 by performing a longest match search comprising one or more routing
7 table queries;
8 a routing table means, coupled to the routing processor means, for providing the
9 routing processor means with a match indication and information
10 regarding a matching routing table entry, if any, of a plurality of routing
11 table entries stored therein in response to a routing table query; and
12 a mask table means, coupled to the routing processor means, for maintaining
13 encoded mask vectors identifying mask lengths of the plurality of routing
14 table entries.

1 30. (New) The packet forwarding device of claim 29, wherein the encoded mask
2 vectors comprise N-bits and are capable of representing N different masks.

1 31. (New) The packet forwarding device of claim 30, wherein the routing table
2 means comprises a Content Addressable Memory (CAM).

1 32. (New) The packet forwarding device of claim 30, wherein the one or more
2 routing table queries are formed by applying a series of masks determined with

3 reference to the mask table means to a search key extracted from the received
4 packet.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

REQUEST FOR APPROVAL TO AMEND DRAWING

Applicants propose correcting Figure 7 to change one symbol. Accordingly, Applicants have included herewith a red-inked original reflected^{ing} the proposed change. No new matter has been added. The Examiner's approval is respectfully requested for the proposed change to Figure 7.

REMARK

Entry of the above-listed amendment is respectfully requested. It is respectfully submitted that no new matter has been introduced by this preliminary amendment.

INVITATION FOR A TELEPHONE INTERVIEW

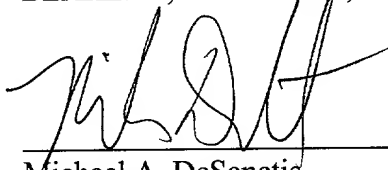
The Examiner is invited to call the undersigned at 408-720-8598 if there remains any issue with the allowance of this case as amended.

CHARGE OUR DEPOSIT ACCOUNT

Please charge any shortage to our Deposit Account No. 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Date: 6/20/01



Michael A. DeSanctis
Reg. No. 39,957

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(303) 740-1980

Amendments -- Marked Up Version

In the Specification:

Presented below are the specification paragraphs, as amended, with changes marked. Insertions are underlined; deletions are bracketed.

Please replace paragraph 0027 on page 10 with the following paragraph:

[0027] Generally speaking, the routing table 230 includes information that indicates on which port (the egress port) a particular address is located. In addition to the specific egress port through which to send the packet, the table might also store information about how the packet should be modified before forwarding. Such changes might include, but not be limited to, a new Media Access Control (MAC) destination address and Virtual LAN (VLAN) address. The routing table 230 typically includes entries having address data and a corresponding payload. The payload may be the forwarding instructions for the particular address or an indication (e.g., an index or a pointer) that can be used to find the forwarding instructions for the particular address. According to one embodiment, the routing table 230 is implemented in whole or part as a Content Addressable Memory (CAM), a random access memory (RAM), such as synchronous RAM (SRAM), or the like that may be implemented using hashing techniques. For example, a hash index generator (not shown), such as a cyclic redundancy checksum (CRC) generator may produce a hash index based on the destination IP address of the packet to be forwarded and the corresponding mask.

Please replace paragraph 0029 on page 11 with the following paragraph:

[0029] Briefly, in operation, the routing process 220 determines appropriate masks to apply to a search key associated with a received packet with reference to the mask table 240 and then queries the routing table 230 to identify the appropriate forwarding instructions for the received packet. For example, the routing process 220 may retrieve an encoded mask vector corresponding to a destination network layer address contained in the received packet (e.g., a source or destination IP address) from the mask table 240 and then perform one or more address look-up requests using those of the masks indicated by the encoded mask vector to have a potential for matching an entry in the routing table 230.

Please replace paragraph 0033 on page 13 with the following paragraph:

[0033] At any rate, using this simplified example, a longest match search will be illustrated for a destination address 0B.01.02.F0 (the search key). In ~~Transmission Control Protocol~~/Internet Protocol (TCP/IP), there might be several entries of a routing table that match a particular address. In this example, network prefix 331 of route (entry) 330, network prefix 341 of route (entry) 340, and network prefix 351 of route (entry) 350 all match 0B.01.02.F0. However, to assure proper delivery of the packet, a network device must use the most specific matching entry, i.e., the entry having the longest mask. Importantly, to be considered a matching entry, the address associated with the entry must match a portion of the search key identified by its mask and the entry's mask length must be less than or equal to the search key's mask length.

Please replace paragraph 0051 on page 19 with the following paragraph:

[0051] At decision block 730, a determination is made whether any further masks are to be attempted. If MaskWord is ~~greater than~~not equal to zero, then at least one more mask remains to be tried and processing continues with processing block 740. However, if MaskWord is equal to zero, then there are no further entries in the routing table 230 that can match the query at any length and longest match search processing is terminated. According to an alternative embodiment, the test for a zero MaskWord is eliminated and instead a special routing table entry is created to match a query of all zeroes and then terminate the search.

Please replace paragraph 0057 on page 21 with the following paragraphs:

[0057] The longest mask described by a MaskWord is found by forming each bit of the mask from the OR of the corresponding bit and all of the lower bits of the MaskWord.

Mask[0] = MaskWord[0];
Mask[1] = | MaskWord[1:0];
Mask[2] = | MaskWord[2:0];
Mask[3] = | MaskWord[3:0];
...
Mask[31] = | MaskWord[31:0];

where "| MaskWord[i,0]" is the REDUCTION OR operation which forms the OR of all bits in the identified range.

This approach of forming the mask employs some very large OR gates for the higher numbered bits. To avoid this people sometimes employ a ripple technique by observing that each bit of the mask already includes the OR of the corresponding bit and all of the lower bits of the MaskWord.

```

Mask[0] = MaskWord[0];
Mask[1] = MaskWord[1] || Mask[0];
Mask[2] = MaskWord[2] || Mask[1];
Mask[3] = MaskWord[3] || Mask[2];
...
Mask[31] = MaskWord[31] || Mask[30];

```

This ripple approach, sometimes called the recursive approach, has the advantage of reducing all of the logic to 31 two-input OR gates and the disadvantage of being slower because it takes 31 gate delays to determine the value of the highest order bit.

A look-ahead method compromises between these extremes of wide gates and long delays by using the ripple approach for most of the bits, but using an occasional wider gate to anticipate what the ripple value will be. Bits higher than the wider gates use their value rather than the ripple value.

```

Mask[0] = MaskWord[0];
Mask[1] = MaskWord[1] || Mask[0];
Mask[2] = MaskWord[2] || Mask[1];
Mask[3] = MaskWord[3] || Mask[2];
Mask[4] = MaskWord[4] || Mask[3];
Mask[5] = MaskWord[5] || Mask[4];

Mask[6] = (MaskWord[6:0]) || Mask[6]; // anticipated value

Mask[7] = MaskWord[7] || Mask[6];
Mask[8] = MaskWord[8] || Mask[7];
Mask[9] = MaskWord[9] || Mask[8];
Mask[10] = MaskWord[10] || Mask[9];
Mask[11] = MaskWord[11] || Mask[10];

Mask[12] = (MaskWord[12:7]) || Mask[6]; // anticipated value

Mask[13] = MaskWord[13] || Mask[12];
Mask[14] = MaskWord[14] || Mask[13];
Mask[15] = MaskWord[15] || Mask[14];
...
Mask[31] = MaskWord[31] || Mask[30];

```

As mentioned earlier, those trying to implement this type of mask expansion in software or in FPGAs cannot efficiently use typical approaches available to ASIC designers. However, processors do have efficient instruction sets and many FPGAs have circuitry “tuned” to implement arithmetic functions. Therefore, if the target platform is software running on a standard processor or an FPGA, the longest mask can be expanded from an encoded mask vector by the following equation, which provides a fast, compact solution:

$$\text{Mask} = ((0 - \text{MaskWord}) | \text{MaskWord})$$

In the Claims:

Presented below are the claims, as amended, with changes marked. Insertions are underlined; deletions are bracketed.

Please amend the claims as follows:

- 1 1. (Unchanged) A method of performing a longest match search comprising:
 - 2 receiving a search key;
 - 3 determining a set of masks that when applied to the search key are known to have
 - 4 a potential for matching an entry in a routing table;
 - 5 forming a routing table query based upon the search key and a longest mask of the
 - 6 set of masks; and
 - 7 applying the routing table query to the routing table.

1 2. (Unchanged) The method of claim 1, further comprising:
2 removing the longest mask from the set of masks; and
3 continuing to apply additional routing table queries until either the set of masks is
4 empty or a matching entry is found in the routing table.

1 3. (Unchanged) The method of claim 1, wherein the search key comprises an
2 Internet Protocol (IP) address.

1 4. (Unchanged) The method of claim 3, wherein the IP address comprises a
2 destination address.

1 5. (Unchanged) The method of claim 3, wherein the IP address comprises a source
2 address.

1 6. (Unchanged) The method of claim 1, wherein said determining a set of masks
2 comprises retrieving an encoded mask vector from a mask table based upon the
3 search key, the encoded mask vector having N bits and capable of identifying N
4 different length masks.

1 7. (Unchanged) The method of claim 1, wherein the longest mask of the set of
2 masks is determined by the following equation: $\text{Mask} = (0 - \text{MaskWord}) |$
3 MaskWord ,
4 where:
5 MaskWord is an encoded mask vector, and
6 Mask is the longest mask identified by MaskWord .

1 8. (Unchanged) A packet forwarding device comprising:
2 a plurality of ports upon which packets are received and transmitted;
3 a routing processor coupled to the plurality of ports to determine an egress port of

4 the plurality of ports for a packet received on an ingress port of the
5 plurality of ports by performing a longest match search comprising one or
6 more routing table queries;
7 a routing table, coupled to the routing processor, to provide the routing processor
8 with a match indication and information regarding a matching routing
9 table entry, if any, of a plurality of routing table entries stored therein in
10 response to a routing table query; and
11 a mask table, coupled to the routing processor, to maintain encoded mask vectors
12 identifying mask lengths of the plurality of routing table entries.

1 9. (Amended) The packet forwarding device of claim 8, ~~the~~wherein encoded mask
2 vectors comprise N-bits and are capable of representing N different masks.

1 10. (Unchanged) The packet forwarding device of claim 8, wherein the routing table
2 comprises a Content Addressable Memory (CAM).

1 11. (Unchanged) The packet forwarding device of claim 8, wherein the one or more
2 routing table queries are formed by applying a series of masks determined with
3 reference to the mask table to a search key extracted from the received packet.

1 12. (Unchanged) A method of forwarding a packet comprising:
2 receiving a packet on an ingress port of a plurality of ports;
3 extracting a destination Internet Protocol (IP) address from a header of the packet;
4 using a portion of the destination IP address to index into a mask table to retrieve
5 an encoded mask vector that identifies a series of masks to be applied to
6 the destination IP address during a longest match search of a routing table,
7 the series of masks representing those masks that are known to have a
8 potential for matching an entry in the routing table when applied to the
9 destination IP address;

10 identifying a longest matching entry in the routing table by performing the longest
11 match search based upon the destination IP address and one or more of the
12 series of masks; and
13 forwarding the packet to a network device associated with the destination IP
14 address via an egress port of the plurality of ports identified by the longest
15 matching entry.

1 13. (Unchanged) The method of claim 12, wherein the portion of the destination IP
2 address comprises the most significant N bits of the destination IP address.

1 14. (Unchanged) The method of claim 12, wherein the encoded mask vector includes
2 a plurality of mask length indicator bits that each indicate a mask length by virtue
3 of their position within the encoded mask vector.

1 15. (Unchanged) The method of claim 12, further comprising updating the mask
2 table to include a new encoded mask vector in response to receiving a new routing
3 table entry.

1 16. (Unchanged) A machine-readable medium having stored thereon data
2 representing sequences of instructions, the sequences of instructions which, when
3 executed by a processor, cause the processor to:
4 receive a search key;
5 determine a set of masks that when applied to the search key are known to have a
6 potential for matching an entry in a routing table;
7 form a routing table query based upon the search key and a longest mask of the
8 set of masks; and
9 apply the routing table query to the routing table.

1 17. (Unchanged) The machine-readable medium of claim 16, wherein the longest
2 mask of the set of masks is determined by the following equation: $\text{Mask} = (0 -$
3 $\text{MaskWord}) \mid \text{MaskWord}$,
4 where:
5 MaskWord is an encoded mask vector, and
6 Mask is the longest mask identified by MaskWord.

1 18. (Unchanged) The machine-readable medium of claim 16, wherein the set of
2 masks is determined by retrieving an encoded mask vector from a mask table
3 based upon the search key, the encoded mask vector having N bits and capable of
4 identifying N different length masks.

19. New

20. New

21. New

22. New

23. New

24. New

25. New

26. New

27. New

28. New

- [illegible]

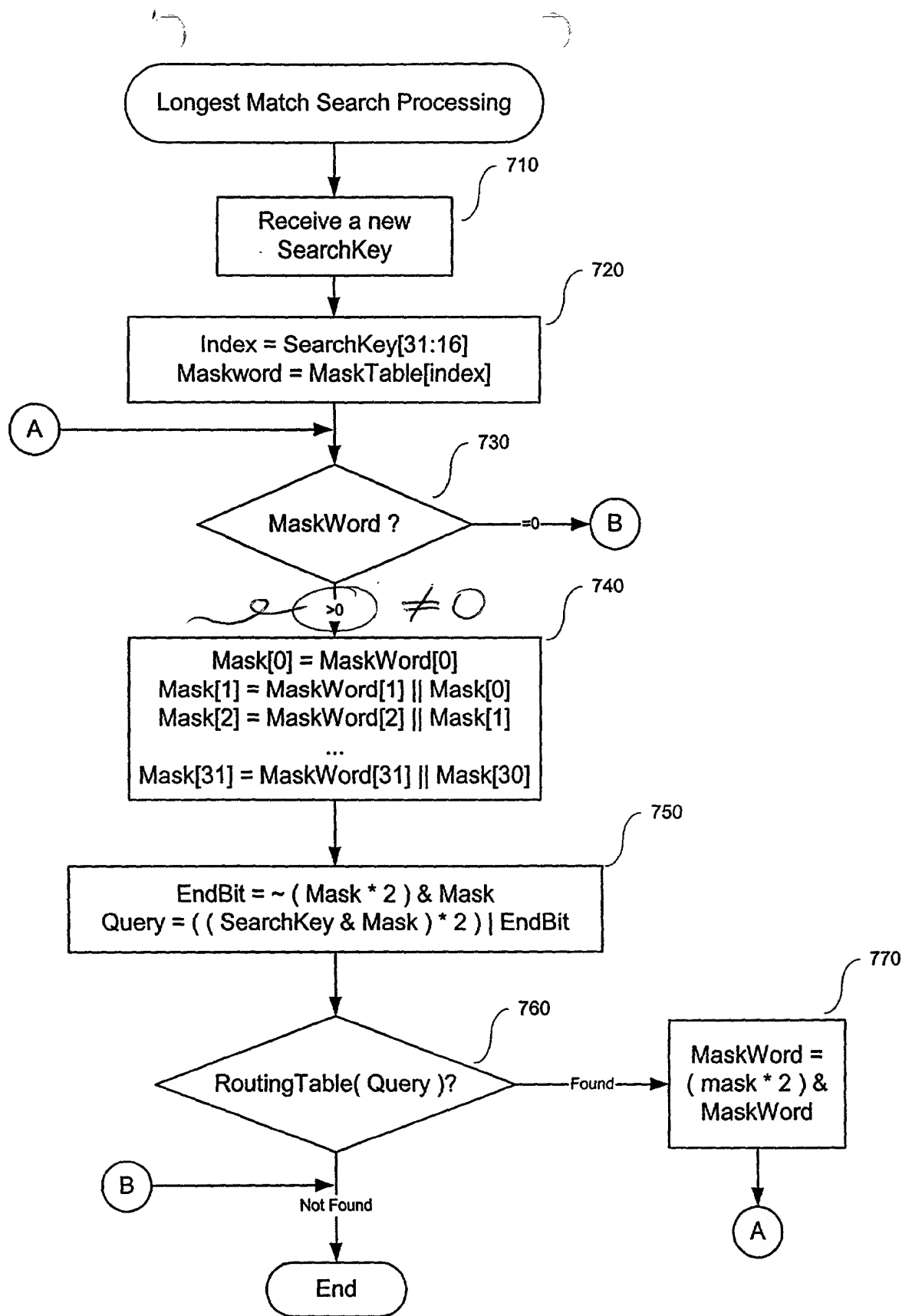


Figure 7